# Data Structures and Algorithms

**Time: 45 Minutes** **Maximum Marks : 20**

## Instructions

- **Make necessary assumptions wherever needed and clearly state the assumptions in your answers.**
- **Necessary steps / justifications / rough work must be present along with the answers**

1. Post order of a binary search tree is given as 5, 27, 20, 48, 62, 55, 30. What is the corresponding Pre-order traversal? If it can not be computed, give the reason. **(1 mark)**

2. An algorithm takes as an input a singly linked list *L* of size n containing *n* integers and calls two functions *f(L)* and *g(L)*. The function *f(L)* finds the maximum number in the linked list *L* and *g(L)* detects whether a cycle exists in *L* or not. What is the time complexity of the algorithm? **(1 mark)**

3. A Ph.D student at NIT Calicut comes up with a new sorting algorithm called "neatSort ". The neatSort is as follows:

```
neatSort(A[], n)
 {
    for(i = 0 to n-1)         // Array starting Index is 0
    {
      j = findMin(A[], i, n)  // It returns the index of minimum element in the subarray A[i:n-1]
     reverse(A[], i, j)       //Reverses the elements in the subarray A[i:j]
    }
 }
```

Note: *findMin* and *reverse* functions are implemented with optimal algorithms.

Example: Input : 3 4 1 5 2

Working of the algorithm:
After the first pass: 1 4 3 5 2
After the second pass: 1 2 5 3 4
After the third pass : 1 2 3 5 4
After the fourth pass : 1 2 3 4 5

a) What is the best case time complexity of the neatSort? Justify your answer. **(1 mark)**

b) Which of the following input is the worst case input for the neatSort. Justify. **(1 mark)**

1) 2 3 5 7 8 6 4 1
2) 1 2 3 4 5 6 7 8
3) 8 7 6 5 4 3 2 1
4) 2 4 6 8 7 5 3 1

4. foo(n)

```
 {
     if(n <=1)          return 1;
     a=0;
     for( i = 1 to n)
               a++;
     return  a + 2*foo(n/2) + foo(n/2);
}
```

a) What is the value of *foo*(8) **(1 mark)**

i) 52 ii) 65 iii) 48 iv) 19

b) Write the recurrence relation for the running time of the function *foo* and Compute the time complexity. Master Theorem can be used. **(2 marks)**

c) What is the space complexity of the above function? **(1 mark)**

5. Consider the following sequence of operations on an empty stack.

push(47); push(46); push(73); pop(); push(78); push(83); pop(); s = pop();

Consider the following sequence of operations on an empty queue.

enqueue(11); enqueue(27); dequeue(); dequeue(); enqueue(46); enqueue(89); q = dequeue();

Compute the value of s+q **(1 mark)**

6. The elements in a max heap are stored in an array as below:

70 50 60 27 35 45 52 18 22 29

What is the parent of element 22? **(1 mark)**

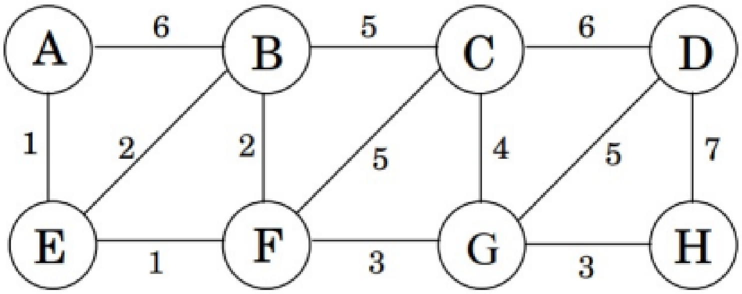
7. What is the time complexity of finding the second smallest element in a max heap of size n. Justify your answer. **(1 mark)**

8. A PhD student doing a literature survey wants to build a software that organizes the research papers. There is a unique identifier for each research paper called *DOI* and it is of 6 digits in length, where each digit is an integer. He decides to use a Hash table as the data structure for storing the details of the research papers, with *DOI* as key. In the implementation of Hash table, quadratic probing is used to resolve the collisions. The $i^{th}$ character of *DOI* is represented as $DOI_i$ . If the hash table size is 907(which is a prime number) then, state for each of the following choice of hash functions whether it is a good choice or a bad choice. Justify your answers.

a) $(DOI_1 + DOI_2 + DOI_3 + DOI_4 + DOI_5 + DOI_6)$ mod 907 **(1 mark)**

b) $(DOI_0 + DOI_1 * 79^1 + DOI_2 * 79^2 + DOI_3 * 79^3 + DOI_4 * 79^4 + DOI_5 * 79^5)$ mod 907 **(1 mark)**

9. Consider the following graph



a) How many minimum spanning trees does it have? **(1 mark)**

b) Suppose Kruskal's algorithm is run on this graph. In what order are the edges added to the MST. **(1 mark)**

c) If DFS is applied on vertex A, compute the number of different possible DFS traversals. **(2 marks)**

10. Consider a variant of the matrix-chain multiplication problem in which the goal is to parenthesize the sequence of matrices so as to maximize, rather than minimize, the number of scalar multiplications. Does this problem exhibit optimal substructure? Explain your answer.

**(1 mark)**

11. You are given an array of distinct integers that is initially decreasing and then increasing. A valley is defined as the element that is smaller than its neighbors. Write an efficient algorithm to find the valley in the given array. Justify its efficiency. **(2 marks)**

*******************************************************************